

Scientific Modelling with Coalgebra–Algebra Homomorphisms

Baltasar Trancón y Widemann
 Ilmenau University of Technology
 baltasar.trancon@tu-ilmenau.de

Michael Hauhs
 University of Bayreuth
 michael.hauhs@uni-bayreuth.de

Draft Revision 1

Abstract

Many recursive functions can be defined elegantly as the unique homomorphisms, between two algebras, two coalgebras, or one each, that are induced by some universal property of a distinguished structure. Besides the well-known applications in recursive functional programming, several basic modes of reasoning about scientific models have been demonstrated to admit such an exact meta-theory. Here we explore the potential of coalgebra–algebra homomorphism that are not a priori unique, for capturing more loosely specifying patterns of scientific modelling. We investigate a pair of dual techniques that leverage (co)monadic structure to obtain reasonable genericity even when no universal properties are given. We show the general applicability of the approach by discussing a suprisingly broad collection of instances from real-world modelling practice.

1 Introduction

This paper explores a (co)algebraic framework for homomorphic and recursive reasoning in and about scientific models; that is, mathematical structures we think in, loaded with interpretations of phenomena from the world we live in. To this end, the paper is structured as follows: the remainder of section 1 reviews the relevant concepts and notations of categorial (co)algebra, with an interlude in section 1.3 that states the motivation and goal more precisely, once terms have been established. Section 2 introduces our dual pair of recursion schemes of interest. Sections 3 and 4 discuss two application domains per scheme. These should be regarded as a set of four related short papers, and are mostly self-contained with respect to discussion and related work. A general conclusion is difficult because of the wide variety of scope, and has to be omitted due to space constraints. We trust the application subsections to speak for themselves. Proofs and illustrations are relegated to the appendix.

1.1 (Co)Algebras of a Functor

Let F be an (endo)functor on a category \mathcal{C} , in all of the following, but without loss of generality, the category of sets. F -algebras are structures $(X, f : FX \rightarrow X)$, where X is called the *carrier* and f the *operation*. F -coalgebras are the dual structures $(X, f : X \rightarrow FX)$. Homomorphisms are morphisms that make a square commute. They can be defined between two algebras or two coalgebras, or between one of each:

$$\begin{array}{ccccc}
 FX & \xrightarrow{f} & X & & X & \xrightarrow{f} & FX & & X & \xrightarrow{f} & FX \\
 Fh \downarrow & & \downarrow h & & h \downarrow & & \downarrow Fh & & Fh \downarrow & & \downarrow h \\
 FY & \xrightarrow{g} & Y & & Y & \xrightarrow{g} & FY & & Y & \xleftarrow{g} & FY
 \end{array}$$

The former two cases of pure (co)algebra homomorphisms have been studied quite extensively in the frameworks of universal (co)algebra. The latter two mixed cases have only relatively recently

received significant attention. The last case, of homomorphisms from an algebra to a coalgebra, remains rather obscure. By contrast, homomorphisms from a coalgebra to an algebra, called coalgebra–algebra homomorphisms and in the following abbreviated as *ca-homomorphisms*, are a very general and expressive framework for divide&conquer schemes of recursion. The common intuition (paraphrased from [5]) is the following: ‘In order to solve (h) a computation problem on complex inputs (X), decompose (f) them into a collection of subproblems (FX), solve these independently (Fh) to obtain a collection of subresults (FY), and compose (g) those to form the final result (Y).’

The composition of two F -(co)algebra homomorphisms is again an F -(co)algebra homomorphism, respectively. The composition of an F -algebra homomorphism with an F -ca-homomorphism, or dually an F -ca-homomorphism with an F -coalgebra homomorphism, is again an F -ca-homomorphism. The following proposition about “new ca-homomorphisms from old ones” is less obvious, but useful for the subsequent discussion.

Lemma 1.1. *Let functor F preserve (co)products. Then the (co)tuplings of ca-homomorphisms are the ca-homomorphisms associated with (co)product-structured (co)algebras, respectively, up to unique isomorphism.*

1.2 Distinguished (Co)Algebras

The F -algebras and F -algebra homomorphisms form a category $\mathbf{Alg}(F)$. Of particular interest are *initial* F -algebras $(\mu F, \text{in})$.¹ For any F -algebra (X, f) there is a unique homomorphism $\llbracket f \rrbracket : (\mu F, \text{in}) \rightarrow (X, f)$, called a *catamorphism*. By Lambek’s lemma, the operation (in) is a bijection. Dually, the F -coalgebras and F -coalgebra homomorphisms form a category $\mathbf{Coalg}(F)$. Of particular interest are *final* F -coalgebras $(\nu F, \text{out})$. For any F -coalgebra (X, f) there is a unique homomorphism $\llbracket f \rrbracket : (X, f) \rightarrow (\nu F, \text{out})$, called an *anamorphism*. The operation (out) is a bijection. For initial algebras and final coalgebras of a variety of well-behaved functors, see [16].

For ca-homomorphisms, the situation is more complex, because they do not compose among themselves. A F -algebra (Y, g) is called *corecursive* [6] if and only if it plays a role analogous to a final F -coalgebra: For any F -coalgebra (X, f) there is a unique F -ca-homomorphism $h : (X, f) \rightarrow (Y, g)$. Dually, a F -coalgebra (X, f) is called *recursive* [5, 12] if and only if it plays a role analogous to an initial F -algebra: For any F -algebra (Y, g) there is a unique F -ca-homomorphism $h : (X, f) \rightarrow (Y, g)$. Trivial examples are obtained by Lambek’s lemma: $(\mu F, \text{in}^{-1})$ is a recursive F -coalgebra; and dually $(\nu F, \text{out}^{-1})$ is a corecursive F -algebra.

1.3 Interlude: Vision

The present paper deals with intermediate situations: Neither are our ca-homomorphisms of interest completely arbitrary, with algebra and coalgebra chosen ad hoc at the same level of particularity; nor is there an obvious candidate for a general (co)recursive (co)algebra, leaving only the partner to be chosen in particular, and the homomorphism induced uniquely. Instead, we investigate situations where either the algebra or the coalgebra part is more central, and the possibility and general shape of homomorphisms is to be studied, independently from its less fixed partner.

This line of investigation ties into our overarching research programme on the recursive nature of scientific modelling posited by Rosen [14]. We shall identify certain modes of abstract formal reasoning over classes of models as (co)algebras, and their rigorous interpretation as “reusable” ca-homomorphisms between a fixed general formal language and a variety of particular models.

In previous work, we have investigated dual, purely algebraic and coalgebraic reasoning modes, and identified them as ‘queries of causality’ and ‘representations of behavior’, respectively [8]. We have also explored various types of mixed modes, and drawn connections between structural operational semantics and cellular automata [18], and between course-of-value iteration and history-dependent dynamics [19], respectively. The present work deals with yet more types of mixed mode, but ones that come as a dual pair, as in [8]. We consider the duality to be of great philosophical interest, although the present modest paper can only give a few first directions.

¹Note the boldface typography that distinguishes fixpoint operator μ and monad multiplication μ .

A great deal of the desired reusability shall be achieved by imposing a (co)monad structure on the functor F . In the following section, we review the relevant concepts and notations very briefly, aware that our presentation can only serve as a glossary, not an introduction.

1.4 (Co)Monads

A *monad* $\mathbb{T} = (T, \eta, \mu)$ is an endofunctor T together with two natural transformations $\eta : 1 \Rightarrow T$ and $\mu : TT \Rightarrow T$, called *unit* and *multiplication*, respectively, such that $\mu \circ \eta T = \text{id}T = \mu \circ T\eta$ and $\mu \circ \mu T = \mu \circ T\mu$. A monad gives rise to a *Kleisli extension* operator that sends any morphism $f : X \rightarrow TY$ to a unique $f^* : TX \rightarrow TY$, such that $\eta^* = \text{id}T$, $f^* \circ \eta_X = f$ and $g^* \circ f^* = (g^* \circ f)^*$; namely $f^* = \mu_Y \circ Tf$ and $f = f^* \circ \eta_X$.

The *Kleisli category* $\mathbf{Kl}(\mathbb{T})$ of a monad has the same objects as the underlying category, but homomorphisms $f : X \xrightarrow{\mathbf{Kl}(\mathbb{T})} Y$ whenever $f : X \rightarrow TY$. Its identity is $\text{id}^{\mathbf{Kl}(\mathbb{T})} = \eta$; composition is given by $g \circ^{\mathbf{Kl}(\mathbb{T})} f = g^* \circ f$. Coalgebras of a monadic functor are of Kleisli type $f : X \xrightarrow{\mathbf{Kl}(\mathbb{T})} X$ and can be iterated. We write $f^{*,n}$ for the morphism such that $f^{*,n} = (f^*)^n$.

For many nice endofunctors F , a *free* monad \mathbb{F}^* can be constructed as follows [3],

$$F^*X = \mu(X + F) \quad F^*(f : X \rightarrow Y) = (\text{in}_Y \circ (f + \text{id}_{FF^*Y}))$$

F^* is a functor, and turns the family of initial algebras into a natural transformation $\text{in} : \text{Id} + FF^* \Rightarrow F^*$. Besides the usual natural transformations that go with a monad, define two additional ones, $\tau : FF^* \Rightarrow F^*$ and $\kappa : F \Rightarrow F^*$ (adapted from [4]), as well as Kleisli extension, simultaneously as:

$$\eta = \text{in} \circ \iota_1 \quad \tau = \text{in} \circ \iota_2 \quad \mu = (\text{id}F^*)^* \quad \kappa = \tau \circ F\eta \quad (f : X \rightarrow F^*Y)^* = \llbracket [f, \tau_Y] \rrbracket$$

Lemma 1.2. *The extra transformations obey the law $\tau = \mu \circ \kappa F^*$.*

Dually, a *comonad* is an endofunctor D together with two natural transformations $\varepsilon : D \Rightarrow 1$ and $\nu : D \Rightarrow DD$, called *counit* and *comultiplication*, respectively, such that $\varepsilon D \circ \nu = \text{id}D = D\varepsilon \circ \nu$ and $\nu D \circ \nu = D\nu \circ \nu$. A comonad gives rise to a *co-Kleisli extension* operator that sends any morphism $f : DX \rightarrow Y$ to a unique $f^\circ : DX \rightarrow DY$, such that $\varepsilon^\circ = \text{id}D$, $\varepsilon_Y \circ f^\circ = f$ and $g^\circ \circ f^\circ = (g \circ f)^\circ$; namely $f^\circ = Df \circ \nu_X$ and $f = \varepsilon_Y \circ f^\circ$.

The *co-Kleisli category* $\mathbf{Cl}(\mathbb{D})$ of a comonad has the same objects as the underlying category, but homomorphisms $f : X \xrightarrow{\mathbf{Cl}(\mathbb{D})} Y$ whenever $f : DX \rightarrow Y$. Its identity is $\text{id}^{\mathbf{Cl}(\mathbb{D})} = \varepsilon$; composition is given by $g \circ^{\mathbf{Cl}(\mathbb{D})} f = g \circ f^\circ$.

A *cofree* comonad F^∞ can often be constructed from an endofunctor F as follows.

$$F^\infty X = \nu(X \times F) \quad F^\infty(f : F^\infty X \rightarrow Y) = \llbracket (f \times \text{id}_{FF^\infty X}) \circ \text{out}_X \rrbracket$$

F^∞ is a functor, and turns the family of final coalgebras into a natural transformation $\text{out} : F^\infty \Rightarrow \text{Id} + FF^\infty$. Besides the usual natural transformations that go with a comonad, define two additional ones, $\theta : F^\infty \Rightarrow FF^\infty$ and $\chi : F^\infty \Rightarrow F$ (dually extrapolated from [4]), as well as co-Kleisli extension, simultaneously as:

$$\varepsilon = \pi_1 \circ \text{out} \quad \theta = \pi_2 \circ \text{out} \quad \nu = (\text{id}F^\infty)^\circ \quad \chi = F\varepsilon \circ \theta \quad (f : F^\infty X \rightarrow Y)^\circ = \llbracket \langle f, \theta_X \rangle \rrbracket$$

Lemma 1.3. *The extra transformations obey the law $\theta = \chi F^\infty \circ \nu$.*

1.5 (Co)Algebras of a (Co)Monad

An algebra (X, f) of a functor T is also an algebra of the monad $\mathbb{T} = (T, \eta, \mu)$, also called an Eilenberg–Moore algebra, if and only if f is compatible with the monad operations, that is $f \circ \eta_X = \text{id}_X$ and $f \circ \mu_X = f \circ Tf$. The algebras of a monad $\mathbf{Alg}(\mathbb{T})$ form a full subcategory of the algebras $\mathbf{Alg}(T)$ of the underlying functor. The monad multiplication is a distinguished algebra operation:

Lemma 1.4. *The \mathbb{T} -algebras of the form (TX, μ_X) are “locally” weakly initial: for every \mathbb{T} -algebra (X, f) , there is a canonical T -algebra homomorphism $f : (TX, \mu_X) \rightarrow (X, f)$. Furthermore, any morphism $h : X \rightarrow Y$ gives rise to a T -algebra homomorphism $Th : (TX, \mu_X) \rightarrow (TY, \mu_Y)$.*

The algebras of a functor F and the algebras of the free monad $\mathbb{F}^* = (F^*, \eta, \mu)$ are in one-to-one correspondence: Each \mathbb{F}^* -algebra (X, f) is of the form $f = e^\sharp = \llbracket \text{id}_X, e \rrbracket$, generated uniquely by the F -algebra (X, e) where $e = f^\flat = f \circ \kappa_X$. Thus we can reformulate Lemma 1.2.

Lemma 1.5. *The free monad extra transformations obey the law $\tau = \mu^\flat$.*

Dually, a coalgebra (X, f) of a functor D is also a coalgebra of the comonad $\mathbb{D} = (D, \varepsilon, \nu)$, if and only if $\varepsilon_X \circ f = \text{id}_X$ and $\nu_X \circ f = Df \circ f$. The coalgebras of a comonad $\mathbf{Coalg}(\mathbb{D})$ form a full subcategory of the coalgebras $\mathbf{Coalg}(D)$ of the underlying functor.

Lemma 1.6. *The \mathbb{D} -coalgebras of the form (DX, ν_X) are “locally” weakly final: for every \mathbb{D} -coalgebra (X, f) , there is the canonical D -coalgebra homomorphism $f : (X, f) \rightarrow (DX, \nu_X)$. Furthermore, any morphism $h : X \rightarrow Y$ gives rise to a D -coalgebra homomorphism $Dh : (DX, \nu_X) \rightarrow (DY, \nu_Y)$.*

The coalgebras of a functor F and the coalgebras of the cofree comonad $\mathbb{F}^\infty = (F^\infty, \varepsilon, \nu)$ are in one-to-one correspondence: Each \mathbb{F}^∞ -coalgebra (X, f) is of the form $f = e^\sharp = \llbracket \langle \text{id}_X, e \rangle \rrbracket$, generated uniquely by the F -coalgebra (X, e) where $e = f^\flat = \chi_X \circ f$.

Lemma 1.7. *The cofree comonad extra transformations obey the law $\theta = \nu^\flat$.*

2 (Co)Kleisli (Co)Induction

We shall demonstrate that well-behaved classes of interesting ca-homomorphisms arise from a pair of dual recursion schemes that leverage (co)monadic structure. The monadic version has been studied recently in the context of trace semantics [4]. Although its dual comonadic twin is straightforward, we are not aware of previous uses; in fact we shall argue why its characteristic properties, albeit essentially useful for our purpose here, are undesirable in many circumstances.

2.1 Kleisli Coinduction

Definition 2.1 (Kleisli Coinduction). Let $\mathbb{T} = (T, \eta, \mu)$ be a monad. Let (X, e) be a T -coalgebra. A morphism $e^\dagger : X \rightarrow TY$ is called a *Kleisli-coinductive solution* of equation e , if and only if $e^\dagger = e^\dagger \circ \mathbf{Kl}(\mathbb{T})_e$.

Lemma 2.2 (Characterization). *Kleisli-coinductive solutions $e^\dagger : X \rightarrow TY$ are precisely the T -ca-homomorphisms to the multiplicative \mathbb{T} -algebra (TY, μ_Y) .*

A Kleisli-coinductive solution $e^\dagger : X \rightarrow TY$ of $e : X \rightarrow TX$ appears as a morphism of type $X \rightarrow Y$ in the Kleisli category. As usual for corecursive functions, its codomain Y is not determined by e at all.

Lemma 2.3 (Substitution). *Let (X, e) be a T -coalgebra and $h : Y \rightarrow Z$ be any morphism. If $e^\dagger : X \rightarrow TY$ is a Kleisli-coinductive solution, then so is $Th \circ e^\dagger : X \rightarrow TZ$.*

Our motivation for studying this particular corecursion scheme is that each Kleisli-coinductive solution extends to many ca-homomorphisms in a regular way, such that its common properties can be studied regardless of the target algebra.

Theorem 2.4 (Universality). *Given any T -coalgebra (X, e) and Kleisli-coinductive solution $e^\dagger : X \rightarrow TY$, there is a canonical family of extensions to T -ca-homomorphisms $(f \circ e^\dagger)$ into all \mathbb{T} -algebras (Y, f) .*

Corollary 2.5 (Recursivity). *If a T -coalgebra (X, e) is recursive, then it has a unique Kleisli-coinductive solution, whose extension is the unique ca-homomorphism.*

Conversely, if a T -coalgebra has a unique Kleisli-coinductive solution, then it is “morally recursive”: there is a unique ca-homomorphism into any \mathbb{T} -algebra that is canonical in the sense of Lemma 2.2. Note that T -algebras which are incompatible with the monadic structure are generally not covered.

On the other hand, Kleisli-coinductive solution can be non-unique, or fail to exist at all. Hence one should regard the equation coalgebra as a *specification* rather than *definition* of ca-homomorphisms. Such a specification can be contradictory or loose; the following proposition shows that it may also be tautological.

Lemma 2.6 (Unit). *Let $\mathbb{T} = (T, \eta, \mu)$ be a monad. For a T -coalgebra of the form (X, η_X) , all morphisms of type $X \rightarrow TY$ are Kleisli-coinductive solutions.*

While a globally tautological specification is evidently of little practical value, we shall make good use of pointwise local tautology in the application section 3.1.

For the special case of a free monad \mathbb{F}^* , we have an asymmetric situation: The F^* -algebras have a distinguished subclass of interest, namely the \mathbb{F}^* -algebras which are equivalent to the simpler class of F -algebras. But F^* -coalgebras generally obey no such constraint of expressivity. If one is imposed deliberately, then a reduced characterization of Kleisli-coinductive solutions in terms of F -ca-homomorphisms can be given.

Definition 2.7 (Basic Equation). Let $\mathbb{F}^* = (F^*, \eta, \mu)$ be a free monad over endofunctor F . An F^* -coalgebra (X, e) is called *basic*, if and only if it factors as $e = \kappa_X \circ e_0$. Note that (X, e_0) is an F -coalgebra.

Lemma 2.8 (Demonadization). *F^* -ca-homomorphisms from basic F^* -coalgebras $(X, \kappa_X \circ e_0)$ to \mathbb{F}^* -algebras (Y, f) are precisely the F -ca-homomorphisms from (X, e_0) to (Y, f^b) .*

Corollary 2.9. *The Kleisli-coinductive solutions $e^\dagger : X \rightarrow F^*Y$ of basic F^* -coalgebras $(X, \kappa_X \circ e_0)$ are precisely the F -ca-homomorphisms from (X, e_0) into the F -algebra (F^*Y, τ_Y) .*

2.2 Co-Kleisli Induction

All of the preceding statements can be dualized.

Definition 2.10 (Co-Kleisli Induction). Let $\mathbb{D} = (D, \varepsilon, \nu)$ be a comonad. Let (Y, k) be a D -algebra. A morphism $k^\dagger : DX \rightarrow Y$ is called a *co-Kleisli-inductive solution* of coequation k , if and only if $k^\dagger = k \circ \text{Cl}(\mathbb{D}) k^\dagger$.

Lemma 2.11 (Characterization). *Co-Kleisli-inductive solutions $k^\dagger : DX \rightarrow Y$ are precisely the ca-homomorphisms from the D -coalgebra (DX, ν_X) .*

Lemma 2.12 (Substitution). *Let (Y, k) be a D -algebra and $h : W \rightarrow X$ be any morphism. If $k^\dagger : DX \rightarrow Y$ is a co-Kleisli-inductive solution, then so is $k^\dagger \circ Dh : DW \rightarrow Y$.*

Theorem 2.13 (Universality). *Given any D -algebra (Y, k) and co-Kleisli-inductive solution $k^\dagger : DX \rightarrow Y$, there are canonical extensions to ca-homomorphisms $(k^\dagger \circ f)$ from all \mathbb{D} -coalgebras (X, f) .*

Corollary 2.14 (Corecursivity). *If a D -algebra (Y, k) is corecursive, then it has a unique co-Kleisli-inductive solution, whose extension is the unique ca-homomorphism.*

Lemma 2.15 (Counit). *Let $\mathbb{D} = (D, \varepsilon, \nu)$ be a comonad. For a D -algebra of the form (Y, ε_Y) , all morphisms of type $DX \rightarrow Y$ are co-Kleisli-inductive solutions.*

Definition 2.16 (Cobasic Coequation). Let $\mathbb{F}^\infty = (F^\infty, \varepsilon, \nu)$ be a cofree comonad over endofunctor F . An F^∞ -algebra (Y, k) is called **cobasic**, if and only if it factors as $k = k_0 \circ \chi_Y$. Note that (Y, k_0) is an F -algebra.

Lemma 2.17 (Decomonadization). *F^∞ -ca-homomorphisms from F^∞ -coalgebras (X, f) to cobasic \mathbb{F}^∞ -algebras $(Y, k_0 \circ \chi_Y)$ are precisely the F -ca-homomorphisms from (X, f^b) to (Y, k_0) .*

Corollary 2.18. *Co-Kleisli-inductive solutions $k^\dagger : F^\infty X \rightarrow Y$ of cobasic F^∞ -algebras are precisely the F -ca-homomorphisms from the F -coalgebra $(F^\infty X, \theta_X)$.*

Our examples in section 4 build on cofree comonads, which can be understood to create spaces of non-well-founded node-labelled trees. In this context, *induction* seems like a thing out of hell: co-Kleisli-inductive solutions are shamelessly allowed to depend on infinite regresses, as well as in vicious circles (via ε) on their own results. But, as we have pointed out, we are happy to sacrifice universal existence and uniqueness of solutions; the scheme is still quite useful as a formal framework for organizing *existing* modes of reasoning, whose potential unsoundness is either of no practical concern, or resolved in domain-specific ways.

3 Applications of Kleisli Coinduction

3.1 Dynamical Systems

Choose a monoid, written additively as $(\Delta, 0, +)$. It may or may not be Abelian, or a group. The (left) product with a monoid, $T = \Delta \times \text{Id}$, is a monad with:

$$\eta(x) = (0, x) \quad \mu(t, (u, x)) = (t + u, x) \quad f^*(t, x) = (t + u, y) \quad \text{where } f(x) = (u, y)$$

The algebras of this monad are exactly the (left) monoid actions; $(X, f : \Delta \times X \rightarrow X)$ where:

$$f(0, x) = x \quad f(t + u, x) = f(t, f(u, x))$$

Monoid actions are the most general formal framework for dynamical systems. The monoid is understood as the structure of durations of time. Default candidates are non-negative/all integers/reals, which model irreversible/reversible discrete/continuous time, respectively. The language that arises from T deals exclusively with the passing of time: In covariant positions, such as the codomain of morphisms in the Kleisli category, $(t, x) \in TX$ reads as “after t time units, x ”; in contravariant positions, such as the domain of T -algebras, $(t, x) \in TX$ reads as “ t time units after x ”. The Kleisli category is the category of sets and timed functions, where $f(x) = (t, y)$ consequently reads as “ f maps x after t time units to y ”. Kleisli composition is sequential in time; delays are accumulated.

Example 3.1 (Harmonic Oscillator). Consider the textbook example of a harmonic oscillator, a point mass m moving frictionlessly along a straight line, acted on by a restoring force proportional, with coefficient k , to its displacement x . In a Newton-style modelling approach, this system is specified by a second-order linear differential equation $\ddot{x} + \frac{k}{m}x = 0$, which simply states that all acceleration is due to the restoring force.

The appropriate analytic model is a dynamical system with $\Delta = \mathbb{R}$ and state vectors $(x, \dot{x}) \in \mathbb{R}^2$.

$$f(t, (x, \dot{x})) = \begin{pmatrix} \cos \omega t & \omega^{-1} \sin \omega t \\ -\omega \sin \omega t & \cos \omega t \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \quad \text{where } \omega = \sqrt{\frac{k}{m}}$$

The ca-homomorphism diagram for this monad takes the equation coalgebra (X, e) as a set X of symbolic states, together with a map e that gives each state $x \in X$ a backwards-looking specification (t, x') , understood as “ t time units after state x' ”. The ca-homomorphisms $h : (X, e) \rightarrow (Y, f)$ are then the instantiations consistent with the forwards-looking dynamics f on a concrete state space Y . Note that partial specifications are included automatically: A state specified tautologically as zero time units after itself is consistent with any instantiation, by pointwise Lemma 2.6.

By unfolding the definition, the Kleisli-coinductive solutions of a T -coalgebra (X, e) are exactly the morphisms of type $e^\dagger : X \rightarrow TY$ such that $e(x) = (t, x')$ and $e^\dagger(x') = (u, y)$ implies $e^\dagger(t + u, y)$. They have a surprisingly rich structure, and admit necessary and/or sufficient conditions, under a variety of mild assumptions concerning the time-likeness of Δ . As outlined before, our framework allows us to study the existence and parameters of solutions, independently of interpretation in a concrete dynamical system.

Definition 3.2. Associate with a T -coalgebra (X, e) several relations:

- $(\sim_e) \subseteq X \times X$ is the reflexive-symmetric-transitive closure of $\pi_2 \circ e$, an equivalence.
- $(\succeq_e) \subseteq X \times X$ is the reflexive-transitive closure of $\pi_2 \circ e$, a preorder.

- Partition into complementary subrelations $(\succ_e) = (\succeq_e) \setminus (\preceq_e)$, a strict partial order, and $(\simeq_e) = (\succeq_e) \cap (\preceq_e) \subseteq (\sim_e)$, an equivalence, as usual.
 - $(\rightsquigarrow_e) \subseteq X \times TX$ is the Kleisli analog of (\succeq_e) , namely $(\rightsquigarrow_e) = \bigcup_{n=0}^{\infty} e^{*,n}$.
- We call (\rightsquigarrow_e) **consistent**, if and only if $x \rightsquigarrow_e (t, x)$ implies $t = 0$.

These relations allow us to state some basic properties of Kleisli-coinductive solutions.

Lemma 3.3. *Let (X, e) be a T -coalgebra. Then it splits into a coproduct of the (\sim_e) -equivalence classes. Furthermore, every Kleisli-coinductive solution $e^\dagger : X \rightarrow TY$ is locally constant in its second component on each (\sim_e) -equivalence class: If $x \sim_e x'$, then $e^\dagger(x) = (u, y)$ and $e^\dagger(x') = (u', y')$ imply $y = y'$.*

Lemma 3.4. *If (\rightsquigarrow_e) is consistent, then for each pair $x \succeq_e x'$ there is a unique t such that $x \rightsquigarrow_e (t, x')$.*

Lemma 3.5. *Let $e^\dagger : X \rightarrow TY$ be a Kleisli-coinductive solution of (X, e) . If $x \rightsquigarrow_e (t, x')$, and $e^\dagger(x') = (u, y)$, then $e^\dagger(x) = (t + u, y)$. In particular, $x \rightsquigarrow_e (0, x')$ implies $e^\dagger(x) = e^\dagger(x')$.*

We say that the monoid $(\Delta, 0, +)$ has right cancellation, if and only if $t + u = u$ implies $t = 0$ for all $t, u \in \Delta$. For cancellative monoids, a reasonable assumption for standard models of time, we can give both a necessary and a sufficient condition on the solvability of a T -coalgebra.

Theorem 3.6. *Assume the monoid $(\Delta, 0, +)$ has right cancellation. Then a T -coalgebra (X, e) has Kleisli-coinductive solutions, only if (\rightsquigarrow_e) is consistent.*

Proof. Assume, for contradiction, that $x \rightsquigarrow_e (t, x)$ where $t \neq 0$. Let $(u, y) = e^\dagger(x)$. Then also $e^\dagger(x) = (t + u, y)$, by Lemma 3.5; hence $t + u = u$, and finally by cancellation $t = 0$. \square

Theorem 3.7. *Assume the monoid $(\Delta, 0, +)$ has right cancellation. Then a T -coalgebra (X, e) has Kleisli-coinductive solutions, if (\rightsquigarrow_e) is consistent, and (\succ_e) is well-founded.*

Proof (Sketch). By well-founded induction and Lemma 3.5. \square

For reversible time, the conditions can be made tight.

Theorem 3.8. *Assume $(\Delta, 0, +)$ is a group. Then a T -coalgebra (X, e) has Kleisli-coinductive solutions, if (and only if) (\rightsquigarrow_e) is consistent.*

Proof (Sketch). Componentwise by Lemma 3.3, then either by reduction to Theorem 3.7 or else by inversion of Lemma 3.4. \square

So far, we have described the solution spaces of T -coalgebras formally, but given no explication of their meaning in the modelling context. We shall now demonstrate that the T -ca-homomorphisms are a very abstract and general account of *time series*: discrete samples of state snapshots over the dynamic evolution of a system.

Fix Δ as either the nonnegative or all reals. Fix a bilaterally infinite sequence of time differences, $(\delta_i \in \mathbb{R})$ for all $i \in \mathbb{Z}$. We write (t_i) for the corresponding partial sums:

$$t_i = + \sum_{j \in [0, i)} \delta_j \quad \text{if } i \geq 0 \qquad t_i = - \sum_{j \in [i, 0)} \delta_j \quad \text{if } i \leq 0$$

Now consider the following T -coalgebras on integer intervals:

$$\begin{array}{lll} X_1 = (-\infty, +\infty) & e_1(i+1) = (\delta_i, i) & (i \in \mathbb{Z}) \\ X_2 = (-\infty, 0] & e_2(i+1) = (\delta_i, i) & (i < 0) \\ X_3 = [0, +\infty) & e_3(0) = (0, 0) & e_3(i+1) = (\delta_i, i) \quad (0 \leq i) \\ X_4 = [0, n] & e_4(0) = (0, 0) & e_4(i+1) = (\delta_i, i) \quad (0 \leq i < n) \end{array}$$

We shall verify that the induced ca-homomorphisms from e_k into dynamical systems are the different shapes of time series: for $k = 1, 2, 3, 4$, the bilaterally infinite, the left-infinite, the right-infinite, and the finite of length $n + 1$, respectively.

In each case, there is only one equivalence class of (\sim_{e_k}) . Hence any Kleisli-coinductive solution $e_k^\dagger : X_k \rightarrow TY$ fixes a single target element $y_0 \in Y$, and is of the form $e_k^\dagger(i) = (t_i + u_0, y_0)$. Consequently, the ca-homomorphisms $h = f \circ e_k^\dagger : (X_k, e_k) \rightarrow (Y, f)$ are of the form: $h(i) = f(t_i + u_0, y_0)$. That is, they are sequences of elements on the trajectory of reference state y_0 , spaced according to the sampling time sequence (t_i) relative to reference time u_0 . Without loss of generality, u_0 can be made to vanish, by rewriting to $h(i) = f(t_i, y'_0)$ where $y'_0 = f(u_0, y_0)$. Furthermore, if (δ_i) is constant, then we obtain the *equidistant* time series, with the simpler form $h(i) = f(i \cdot \delta, y_0)$.

Example 3.9 (Period). For the harmonic oscillator from Example 3.1, the well-known periodic behaviour can be captured very concisely and naturally, by stating that the bilaterally infinite, equidistant time series specification coalgebra (X_1, e_1) with $\delta = 2\pi\omega^{-1}$ admits only constant ca-homomorphisms.

Example 3.10 (Zeno). As a sidenote, the pathological class of non-equidistant, right-infinite but bounded time series have played an important, puzzling role in ancient Greek “scientific modelling”. For instance, Zeno’s stadium run can be specified by T -coalgebra (X_3, e_3) with $\delta_i = 2^{-(i+1)}$, where half of the remaining time is consumed at each step.

This account of discrete time series over continuous dynamical systems may seem a little contrived, simply because standard textbook presentations look fairly different, and do not suggest the use of a formal framework beyond simple set theory and “index magic”. The following example shall serve, among other things, to counter that impression; there the study of ca-homomorphisms from Kleisli coinduction leads to observations and structures that are completely standard in the field.

3.2 Markov Chains

Our second example application, although conceptually rather a little more advanced, can be presented with much less technical detail, because we can build on standard accounts in terms of far more expressive mathematics; see for instance [9].

Consider discrete distributions on a set X , with possibly countably infinite support, $TX = \{\pi : X \rightarrow [0, 1] \mid \sum_{x \in X} \pi(x) = 1\}$. The distribution space TX has a convex structure. Hence its elements can be understood as formal sums $\sum_i p_i x_i$, and the operations

$$Tf(\sum_i p_i x_i) = \sum_i p_i f(x_i) \quad \eta(x) = 1x \quad \mu(\sum_i p_i (\sum_j q_{ij} x_{ij})) = \sum_{ij} p_i q_{ij} x_{ij}$$

define a monad $\mathbb{T} = (T, \eta, \mu)$ [10]. Alternatively, the elements $\pi \in TX$ can be understood as stochastic row vectors Π with columns index by X . By extension, Kleisli morphisms of type $f : X \rightarrow TY$ can be understood as right stochastic matrices F of shape $X \times Y$, by currying.

Lemma 3.11. *Kleisli extension and composition are right matrix multiplication:*

$$f^*(\pi) = \Pi F \quad g \circ^{\text{Kl}(\mathbb{T})} f = FG$$

The T -coalgebras (X, e) are exactly the *space-discrete, time-homogeneous Markov chains*, a model class pervading almost all corners of science. The \mathbb{T} -algebras are *convex sets*: structures (Y, f) with a set Y closed under a given interpretation f of formal convex combinations. In a very abstract but logically precise sense, the corresponding ca-homomorphisms employ convex sets as models of the *long-term behaviour* of a Markov chain: Let Y be a set of possible long-term behaviours (refraining from jumping to the conclusion of well-known candidates). A ca-homomorphism is a map $h : X \rightarrow Y$ of Markov states to long-term behaviours that is consistent with transition structure e , namely $e(x) = \sum_i p_i x'_i$ implies $h(x) = f(\sum_i p_i h(x'_i))$. That is, the long-term behaviour associated with a state is an appropriately weighted convex combination of the long-term behaviours of its successors.

The default candidates for long-term behavior that we have alluded to are of course the *stationary distributions*, the fixpoints of f^* . That our ca-homomorphisms are a generalization, is apparent from the Kleisli-coinductive solutions: the matrix version by Lemma 3.11 of the coinduction property gives the linear fixpoint equation $E^\dagger = E^\dagger E$, such that the rows of E^\dagger are logically independent unitary left eigenvectors of E , which is the standard characterisation of stationary distributions.

Note that, even though the intuition of distributions represented as formal convex combinations of states is quite dominant, the target space Y is completely abstract in this framework; any convex set will do from an axiomatic perspective. The classical presentation can be retrieved as the distinguished target \mathbb{T} -algebra (TX, μ_X) .

Regarding the degrees of freedom in ca-homomorphisms , Markov chains are an interesting escalation over exact dynamical systems: On the one hand, they support an analogous concept of coproduct structure, namely the so-called *communicating classes*, which are the strongly connected components of the non-zero transition probability graph. On the other hand, they have a well-known additional degree of freedom: even irreducible (totally connected) Markov chains can have ambiguous stationary distribution, if they are *periodic*; that is, some state has zero transition probability to itself in n steps, for arbitrarily large n .

Note that, for subtle interpretative reasons, the direction of time in this subsection is reversed with respect to the previous one; the language of equation coalgebras speaks about the future rather than the past, whereas the algebra constructs the past rather than the future; in philosophical terms, the latter is governed by *final* rather than *causal* reasons.

4 Applications of Co-Kleisli Induction

4.1 Economic Games

As the first application of co-Kleisli induction, we review the coalgebraic presentation of economic game theory from [1]. They construct perfect information games in extensional form by means of anamorphisms into the final coalgebra of a suitable game functor. We adapt their formalization slightly.

Definition 4.1 (Game Functor). Fix a set \mathcal{A} of *agents* and a set \mathcal{C} of *choices*. The game functor is defined as follows:

$$GX = \mathbb{R}^{\mathcal{A}} + \mathcal{A} \times X^{\mathcal{C}}$$

Elements are of either of the forms $\iota_1(u : \mathcal{A} \rightarrow \mathbb{R})$ or $\iota_2(a, m : \mathcal{C} \rightarrow X)$. They are understood as stating, respectively, that ‘the game terminates with payoff $u(x)$ for all agents $x \in \mathcal{A}$ ’, or that ‘the game continues with agent a ’s turn and continuation $m(c)$ for any possible choice $c \in \mathcal{C}$ that a can make’.

In [1], possibly infinite game trees are unfolded from coalgebras (X, e) that model games as transition systems. The concrete states from X are erased by the anamorphism as usual, only the transition structure is retained. By contrast, we shall consider the cofree comonad G^∞ , and the lifted G^∞ -coalgebras (X, e^\sharp) . In comparison, they can be understood as running ‘with logging’; the original states are retained as node labels in the tree. This allows for concise treatment of game tree evaluations as node label processors.

We shall present only a single, very well-known evaluation algebra, in order to demonstrate the applicability of our approach. The field of game theory is broad and deep, and in great need of formal frameworks that give concise and elegant notations. Thus we mark the extension of the present sketch as an interesting topic for future research, which however requires a great deal more effort and space.

One of the historically and logically most basic evaluation techniques for games is *backward induction* [22].

The typical application of backward induction, especially in economic uses of game theory such as discussion in [1], is the computation of expected payoff, under the assumption that all agents choose such as to maximize their own payoff. This notion can be stated extremely concisely as the generating G -algebra $(\mathbb{R}^{\mathcal{A}}, k_0)$ of a cobasic G^∞ -algebra $(\mathbb{R}^{\mathcal{A}}, k = k_0 \circ \chi_{\mathbb{R}^{\mathcal{A}}})$:

$$k_0(\iota_1(u)) = u \qquad k_0(\iota_2(a, m)) = m(\arg \max c. m(c)(a))$$

It follows that any co-Kleisli-inductive solution $k^\dagger : G^\infty X \rightarrow \mathbb{R}^{\mathcal{A}}$ satisfies $k^\dagger(g) = u$ if $\theta_X(g) = \iota_1(u)$, and the fairly convoluted $(k^\dagger \circ m)(\arg \max c. (k^\dagger \circ m)(c)(a))$ if $\theta_X(g) = \iota_2(a, m)$, which is on second thought a natural formalization of the above prose description for game trees in general,

regardless of the game being played. The canonical extension to a ca-homomorphism from a concrete game coalgebra (X, f) , namely $k^\sharp \circ f$, restricts the evaluation to legal trees with a given starting state.

It is well-known and appreciated in game theory that there is a unique solution, if only finite games are considered. In our framework, that is to say that finite games can be accommodated by the *recursive subcomonad* of G^∞ [21]. We conjecture that some standard techniques for forcing canonical solutions in infinite cases, such as *discounting*, can be expressed conveniently in terms of a non-cobasic evaluation algebra.

As a sidenote, the very term *backward induction* embodies the characteristic difficulty of writing about scientific modelling, namely the required double reading with regard to formal representation and phenomenological interpretation, respectively. As we have noted in the preceding section, the direction of time, and hence the word ‘backward’ is a concept of the phenomenological dimension; by contrast the algebraic structure and hence the word ‘induction’ is a concept of the formal dimension, making the composite term essentially a category mistake.

4.2 Lindenmayer Fractals

The following, last example application, while another instance of co-Kleisli-induction for a fixed algebra, has added value in two respects: Firstly, it demonstrates the definition of a domain-specific comonad as part of the formal modelling framework. Secondly and paradoxically, it makes perfectly reasonable use of essentially non-well-founded induction.

It is a well-known statement, and frequent subject of mathematical instruction, that certain fractal shapes can be specified by Lindenmayer systems (L-systems), a formalism akin to Chomsky grammars that has originally been developed for formal modelling of the growth behavior of plants [11, 13, 15]. This statement, while morally true, is to be taken with a pinch of salt; considerable effort is required for a satisfactorily rigorous explication. We shall demonstrate that the framework of ca-homomorphisms is a natural and expressive background for this line of reasoning. Confer [7] for another account of fractals that builds on ca-homomorphisms (their diagram 3), but proceeds in an interestingly different way.

Definition 4.2 (Curve). Fix some real Banach space V , say without loss of generality $V = \mathbb{R}^2$. Fix a real interval I . An I -curve is a continuous map $f : I \rightarrow V$.

Definition 4.3 (Unit Curve). Fix some unit vector $\vec{e} \in V$ with $\vec{e} \neq \vec{0}$, say without loss of generality $\vec{e} = (1, 0)$. An \vec{e} -unit curve is a $[0, 1]$ -curve f with $f(0) = \vec{0}$ and $f(1) = \vec{e}$. We write C_1 for the set of all unit curves.

Example 4.4. The simplest unit curve is the *stroke*: $str(x) = x\vec{e}$.

Definition 4.5 (Context-Free L-System). As observed in [20], simple context-free L-systems, with nonterminal symbols only, are finite coalgebras of the (nonempty) list functor $\mathcal{L}_0 X = X^+$: The carrier is the set of nonterminals, and the operation is simply the set of production rules. The traditional semantics of an L-system (X, f) is the sequence of iterations $(f^{*,n}(s))$, seeded with a start symbol $s \in X$.

We write $\langle \rangle$ for the empty list, and $x :: r$ for a list with first element x and rest list r , as in the programming language ML. Complex lists $x_1 :: \dots :: x_n :: \langle \rangle$ are condensed to $\langle x_1, \dots, x_n \rangle$. The operator \oplus denotes binary concatenation of lists.

Definition 4.6 (Turn). Fix some group (G, \cdot, i) of linear isometries on V , here the special orthogonal group of rotations $\rho_\alpha = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$ parameterized by angle α . The elements of G are called *turns*. The action on vectors is written $\bullet : G \times V \rightarrow V$.

Definition 4.7 (Fractal L-System). L-systems for specifying fractals are extensions of simple context-free L-systems, that is, finite coalgebras of a more complex functor. In particular, they come with an additional real-valued shrink factor as a per-rule attribute, as well as terminal symbols for turns. As hinted in the short paper [20], these can be expressed conveniently in terms of composition $\mathcal{L}_{\text{fr}} = \mathcal{S}\mathcal{L}_0\mathcal{T}$ with additional functors $\mathcal{S} = \mathbb{R} \times \text{Id}$ and $\mathcal{T} = G + \text{Id}$, respectively. Note that \mathcal{L}_{fr} is monotonic and even preserves inclusions.

Example 4.8 (Koch Curve). The following singleton L-system specifies a fractal shape known as the Koch curve. We use a visually evocative, short notation where the shrink factor appears as superscript to the production rule arrow, and nonterminals and turns, represented by their angles, are simply juxtaposed.

$$K \xrightarrow{3} K (+\alpha) K (-2\alpha) K (+\alpha) K$$

This contains the same information as the more formal, but also much more cumbersome full notation of $f : \{K\} \rightarrow \mathcal{L}_{\text{fr}}\{K\}$ as

$$f(K) = (3, \langle \iota_2(K), \iota_1(r_{+\alpha}), \iota_2(K), \iota_1(r_{-2\alpha}), \iota_2(K), \iota_1(r_{+\alpha}), \iota_2(K) \rangle)$$

In order for an L-system to meaningfully specify a fractal curve, additional constraints must be met. These are not readily expressed in syntactic terms, and because of the inherently self-referential nature of L-systems, great care must be exercised not to state them in a circular way.

Definition 4.9 (Well-Formed Rule). We give an abstract geometric interpretation of expressions of type $\mathcal{L}_{\text{fr}}X$, under the assumption that each nonterminal represents a unit curve. To this end, we track the overall change of position and orientation effected by sequentially tracing curves and executing turns, as in ‘turtle’ graphics, disregarding the shrink factor.

$$\begin{aligned} \text{span} : \mathcal{L}_0\mathcal{T}X \rightarrow V & & \text{dir} : \mathcal{L}_0\mathcal{T}X \rightarrow G \\ \text{span}(t) = \begin{cases} \vec{0} & \text{if } t = \langle \rangle \\ \rho \bullet \text{span}(u) & \text{if } t = \iota_1(\rho) :: u \\ \vec{e} + \text{span}(u) & \text{if } t = \iota_2(x) :: u \end{cases} & & \text{dir}(t) = \begin{cases} i & \text{if } t = \langle \rangle \\ \rho \cdot \text{dir}(u) & \text{if } t = \iota_1(\rho) :: u \\ \text{dir}(u) & \text{if } t = \iota_2(x) :: u \end{cases} \end{aligned}$$

A rule is called *well-formed* if and only if it is properly shrinking, spans the unit vector when shrunk, and has balanced turns: $\text{wfr}((a, t)) \iff a > 1 \wedge \text{span}(t) = a\vec{e} \wedge \text{dir}(t) = i$.

Lemma 4.10. *The abstract interpretation is additive.*

$$\text{span}(t \oplus u) = \text{span}(t) + \text{dir}(t) \bullet \text{span}(u) \quad \text{dir}(t \oplus u) = \text{dir}(t) \cdot \text{dir}(u)$$

Lemma 4.11. *Well-formedness is additive: $\text{wfr}((a, t)) \wedge \text{wfr}((b, u)) \implies \text{wfr}((a + b, t \oplus u))$.*

Lemma 4.12. *Well-formedness is invariant under the fractal Lindenmayer functor.*

This result allows us to abstract from the contents of a rule, retaining only the shape, and to construct a subfunctor for well-formed rules.

Definition 4.13 (Rule Shape). Let $1 = \{*\}$ be the final object, with unique morphisms $!_X : X \rightarrow 1$. We call $\mathcal{L}_{\text{fr}}1$ the *shape space*, and $\mathcal{L}_{\text{fr}}!_X : \mathcal{L}_{\text{fr}}X \rightarrow \mathcal{L}_{\text{fr}}1$ the *shape map*.

Lemma 4.14. *A rule is well-formed if and only if its shape is.*

Definition 4.15 (Well-Formed Fractal L-System). We obtain a subfunctor $\mathcal{L}_{\text{wfr}} = \mathcal{L}_{\text{fr}}|_{\text{wfr}}$ of the general fractal Lindenmayer functor by restriction to well-formed rules.

Definition 4.16 (Rule Interpretation). The concrete interpretation of rules differs from the abstract interpretation with *span* by actually composing the subcurves listed in a rule, concatenating their domains. We model this idea as a \mathcal{L}_{fr} -algebra (Y, draw) over the general function space $Y = V^{[0,1]}$, not assuming continuity for now.

In order to obtain the shape of a unit curve, both the domain and the result need to be compressed, according to the length of the rule and the shrink factor, respectively. In the following, let ℓ denote the length of list $t \in \mathcal{L}_0\mathcal{T}Y$.

$$\begin{aligned} \text{draw}((a, t))(z) &= a^{-1} \text{step}(t)(\ell z) \quad \text{where} \\ \text{step}(t)(x) &= \begin{cases} \vec{0} & \text{if } z \in [0, 1] & t = \iota_1(\rho) :: u & \text{(A)} \\ \vec{0} & \text{if } z = 0 & t = \iota_2(f) :: u & \text{(B)} \\ f(x) & \text{if } z \in (0, 1) & t = \iota_2(f) :: u & \text{(C)} \\ \vec{e} & \text{if } z = 1 & t = \iota_2(f) :: u & \text{(D)} \\ \rho \bullet \text{step}(u)(z - 1) & \text{if } z > 1 & t = \iota_1(\rho) :: u & \text{(E)} \\ \vec{e} + \text{step}(u)(z - 1) & \text{if } z > 1 & t = \iota_2(f) :: u & \text{(F)} \end{cases} \end{aligned}$$

The auxiliary function *step* has dependent type; for every argument t of length $\ell > 0$, it yields a function of type $V^{[0, \ell]}$. Note that the case $t = \langle \rangle$ does not occur, even recursively.

By overlapping the subcurves at their glueing points in the composition, and even sending turns to constant points, we depart radically from the way of [7], where *injectivity* is a major theoretical goal, and successfully solved technical obstacle. Whereas they have been concerned with faithful *representation* of fractal space, our approach is more a *computational* account: By virtue of the piecewise constancy introduced by turns, a point $f(x)$ for random x can be computed *exactly* by finite recursion *almost surely*, except for some pathological L-systems. We discuss this property more deeply in a forthcoming companion paper.

Lemma 4.17. *The endpoint of step is predicted by span: $\text{step}(t)(\ell) = \text{span}(t)$.*

Theorem 4.18. *Interpretation of well-formed rules preserves unit curves: $\text{draw}(r) \in C_1$ for all $r \in \mathcal{L}_{\text{wfr}}(C_1)$.*

Proof (Sketch). By piecewise continuity with agreement at glueing points. \square

Definition 4.19 (Well-Formed Rule Interpretation). We obtain an \mathcal{L}_{wfr} -algebra $(C_1, \text{draw}_{\text{wfr}})$ by restricting rule interpretation *draw* to unit curves.

Definition 4.20. Let $\alpha : \mathcal{L}_{(\text{w})\text{fr}} \Rightarrow \mathcal{P}$ be the obvious natural transformation that extracts the set of nonterminal symbols occurring in a rule. Let $\mathbb{P} = (\mathcal{P}_\omega, \eta, \mu)$ be the well-known *finite* powerset monad. Now consider the cofree comonad over \mathcal{L}_{wfr} . The object $\mathcal{L}_{\text{wfr}}^\infty X$ can be considered as the space of non-well-founded X -labelled trees of nested well-formed rules. From these we can extract sets of nestedly occurring rule shapes: The natural transformation

- $\rho = \eta \circ \chi : \mathcal{L}_{\text{wfr}}^\infty \Rightarrow \mathcal{P}_\omega \mathcal{L}_{\text{wfr}}$ extracts the single root shape;
- $\sigma = \alpha F^\infty \circ \theta : \mathcal{L}_{\text{wfr}}^\infty \Rightarrow \mathcal{P}_\omega \mathcal{L}_{\text{wfr}}^\infty$ extracts the immediate subtrees.

Thus the extraction of all shapes occurring at some finite depth in $t \in \mathcal{L}_{\text{wfr}}^\infty X$ is defined as $S_X(t) = \bigcup_{n=0}^\infty (\mathcal{P}_\omega !_X \circ \rho_X^* \circ \sigma_X^{*,n})(t)$, where Kleisli extension is over \mathbb{P} . Confer the relation (\leadsto_e) from Definition 3.2.

Definition 4.21. We obtain the *shape-finitary* subcomonad \mathbb{D} of the cofree comonad of well-formed L-systems: $DX = \{t \in \mathcal{L}_{\text{wfr}}^\infty X \mid S_X(t) \text{ finite}\}$. Clearly, all actual such L-systems (X, f) , which are *finite* \mathcal{L}_{wfr} -coalgebras, extend to \mathbb{D} -coalgebras (X, f^\sharp) . Note the analogy to *rational coalgebras* [2, 17].

Finally we construct an interpretation (coequation) algebra, the very goal of this whole subsection, by extending draw_{wfr} to a cobasic $\mathcal{L}_{\text{wfr}}^\infty$ -algebra, and restricting its domain to shape-finitary trees: $k = \text{draw}_{\text{wfr}} \circ \chi_{C_1}|_{DC_1}$.

The ca-homomorphisms $h : (X, f) \rightarrow (C_1, k)$ map nonterminal symbols of L-systems to unit curves, with the obvious self-similar geometric consistency condition. Now we can make the point of our nontrivial efforts and restrictions.

Theorem 4.22. *The coequation D-algebra (C_1, k) is corecursive; a unique ca-homomorphism exists from any \mathbb{D} -coalgebra.*

Proof (Sketch). By uniform continuity and uniqueness on the dense subdomain of recursive glueing points. \square

Note that, while the comonadic structure appears not to be used by the cobasic algebra, it is required for expressing the restriction to shape-finitary L-systems, which in turn is a necessary condition for our proof; infinitary counterexamples can be conceived.

References

- [1] S. Abramsky and V. Winschel. Coalgebraic analysis of subgame-perfect equilibria in infinite games without discounting. *arXiv*, 1210.4537, 2012.
- [2] J. Adámek, S. Milius, and J. Velebil. Iterative algebras at work. *Math. Structures Comput. Sci.*, 16:1085–1131, 2006.
- [3] M. Barr. Coequalizers and free triples. *Math Z.*, 116:307–322, 1970.
- [4] F. Bonchi, S. Milius, A. Silva, and F. Zanasi. How to kill epsilons with a dagger: A coalgebraic take on systems with algebraic label structure. In *Proc. CMCS*, volume 8446 of *LNCS*, pages 53–74, 2014.
- [5] V. Capretta, T. Uustalu, and V. Vene. Recursive coalgebras from comonads. *ENTCS*, 106:43–61, 2004.
- [6] V. Capretta, T. Uustalu, and V. Vene. Corecursive algebras: A study of general structured corecursion. In *Proc. SBMF*, volume 5902 of *LNCS*, pages 84–100, 2009.
- [7] I. Hasuo, B. Jacobs, and M. Niqui. Coalgebraic representation theory of fractals. *ENTCS*, 265:351–368, 2010.
- [8] M. Hauhs and B. Trancón y Widemann. Applications of algebra and coalgebra in scientific modelling, illustrated with the logistic map. *ENTCS*, 264(2):105–123, 2010.
- [9] R. A. Howard. *Dynamic Probabilistic Systems*, volume 1. Wiley & Sons, 1971.
- [10] B. Jacobs. Convexity, duality and effects. In *Proc. TCS*, volume 323 of *IFIP AICT*, pages 1–19, 2010.
- [11] A. Lindenmayer. Mathematical models for cellular interaction in development. *J. Theoret. Biology*, 18:280–315, 1968.
- [12] G. Osius. Categorical set theory: A characterization of the category of sets. *J. Pure and Applied Algebra*, 4(1):79–119, 1974.
- [13] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, 1990.
- [14] R. Rosen. *Life Itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life*. Complexity in Ecological Systems. Columbia University Press, 1991.
- [15] G. Rozenberg and A. Salomaa. *The Mathematical Theory of L-Systems*. Academic Press, 1980.
- [16] J. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.
- [17] B. Trancón y Widemann. $V \rightarrow M$: A virtual machine for strict evaluation of (co)recursive functions. In *Proc. IFL*, volume 3474 of *LNCS*, pages 90–107. Springer-Verlag, 2004.
- [18] B. Trancón y Widemann and M. Hauhs. Distributive-law semantics for cellular automata and agent-based models. In *Proc. CALCO*, volume 6859 of *LNCS*, pages 344–358, 2011.
- [19] B. Trancón y Widemann and M. Hauhs. Algebraic–coalgebraic recursion theory of history-dependent dynamical system models. In *Proc. CMCS*, volume 8446 of *LNCS*, pages 225–244, 2014.
- [20] B. Trancón y Widemann and J. Winter. Lindenmayer systems, coalgebraically. In *Draft Proc. CMCS*. Tallinn University of Technology, 2012. Short contribution.
- [21] T. Uustalu and V. Vene. The recursion scheme from the cofree recursive comonad. *ENTCS*, 229(5):135–157, 2011.
- [22] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

A Omitted Proofs

This section gives or completes all proofs that we consider informative or non-elementary. Some boring immediate or inductive proofs are omitted altogether.

Proof of Lemma 1.1. Let F preserve coproducts. Let index i range over some set I in all of the following. Then, for any family of objects (X_i) , there is a unique isomorphism $j : \coprod_i F X_i \rightarrow F \coprod_i X_i$ such that, for any family of morphisms $(h_i : X_i \rightarrow Y)$, we have $[F h_i] = F[h_i] \circ j$. Now let (X_i, f_i) be a family of F -coalgebras. Then $(X, f) = (\coprod_i X_i, j \circ \coprod_i f_i)$ is again an F -coalgebra. Let (Y, g) be a fixed F -algebra. The ca-homomorphisms of type $h : (X, f) \rightarrow (Y, g)$ are in one-to-one correspondence with the families of morphisms (h_i) where all members are ca-homomorphisms $h_i : (X_i, f_i) \rightarrow (Y, g)$, namely $h = [h_i]$:

$$\begin{array}{ccccc}
 \coprod_i X_i & \xrightarrow{\coprod_i f_i} & \coprod_i F X_i & \xrightarrow{j} & F \coprod_i X_i \\
 \downarrow [h_i] & \swarrow \iota_i & \downarrow \iota_i & \downarrow [F h_i] & \downarrow F[h_i] \\
 & X_i & \xrightarrow{f_i} & F X_i & \\
 & \downarrow h_i & \searrow F h_i & & \\
 Y & \xleftarrow{g} & F Y & &
 \end{array}$$

For products of algebras dually. □

Proof of Lemma 1.2 & 1.3. By natural laws of coproduct and monad, and catamorphic μ .

$$\begin{array}{ccc}
 & F F^* & \\
 & \downarrow F \eta F^* & \\
 & F F^* F^* & \xrightarrow{F \mu} F F^* \\
 \uparrow \kappa F^* & \downarrow \iota_2 & \downarrow \iota_2 \\
 F^* + F F^* F^* & \xrightarrow{F^* + F \mu} & F^* + F F^* \\
 \downarrow \text{in } F^* & & \downarrow [\text{id } F^*, \tau] \\
 F^* F^* & \xrightarrow{\mu} & F^*
 \end{array}$$

For θ dually. □

Proof of Lemma 1.4 & 1.6. The former clause by compatibility of f with μ ; the latter clause by naturality of μ .

$$\begin{array}{ccc}
 T T X & \xrightarrow{\mu_X} & T X \\
 T f \downarrow & & \downarrow f \\
 T X & \xrightarrow{f} & X
 \end{array}
 \qquad
 \begin{array}{ccc}
 T T X & \xrightarrow{\mu_X} & T X \\
 T T h \downarrow & & \downarrow T h \\
 T T Y & \xrightarrow{\mu_Y} & T Y
 \end{array}$$

For ν dually. □

Proof of Lemma 2.2 & 2.11. Proposition laid out in underlying category.

$$\begin{array}{ccc}
 T X & \xleftarrow{e} & X \\
 T e^\dagger \downarrow & \searrow (e^\dagger)^* & \downarrow e^\dagger \\
 T T Y & \xrightarrow{\mu_Y} & T Y
 \end{array}$$

For k^\dagger dually. □

Proof of Lemma 2.3 & 2.12. By composition of Lemma 1.4, second clause, with Lemma 2.2.

$$\begin{array}{ccc}
TX & \xleftarrow{e} & X \\
Te^\dagger \downarrow & \searrow (e^\dagger)^* & \downarrow e^\dagger \\
TTY & \xrightarrow{\mu_Y} & TY \\
TTh \downarrow & & \downarrow Th \\
TTZ & \xrightarrow{\mu_Z} & TZ
\end{array}$$

For k^\dagger dually. □

Proof of Theorem 2.4 & 2.13. By composition of Lemma 1.4, first clause, with Lemma 2.2.

$$\begin{array}{ccc}
TX & \xleftarrow{e} & X \\
Te^\dagger \downarrow & \searrow (e^\dagger)^* & \downarrow e^\dagger \\
TTY & \xrightarrow{\mu_Y} & TY \\
Tf \downarrow & & \downarrow f \\
TY & \xrightarrow{f} & Y
\end{array}$$

For k^\dagger dually. □

Proof of Lemma 2.8 & 2.17. By naturality of κ .

$$\begin{array}{ccccc}
& & e_0 & & \\
FX & \xrightarrow{\kappa_X} & F^*X & \xleftarrow{e} & X \\
Fh \downarrow & & F^*h \downarrow & & \downarrow h \\
FY & \xrightarrow{\kappa_Y} & F^*Y & \xrightarrow{f} & Y \\
& & f^b & &
\end{array}$$

For χ dually. □

Proof of Lemma 3.4. Assume that $x \rightsquigarrow_e (t, x')$ and $x \rightsquigarrow_e (u, x')$. Then there are m, n such that $e^{\star, m}(x) = (t, x')$ and $e^{\star, n}(x) = (u, x')$. Assume without loss of generality $m < n$. Then

$$\begin{aligned}
(u, x') &= e^{\star, n}(x) \\
&= ((e^\star)^{n-m} \circ e^{\star, m})(x) \\
&= (e^\star)^{n-m}((t, x')) \\
&= (e^{\star, n-m})^\star((t, x')) \\
&= (t + v, x') \quad \text{where } e^{\star, n-m}(x') = (v, x') \\
&\quad \quad \quad x' \rightsquigarrow_e (v, x')
\end{aligned}$$

With $v = 0$ by consistency, we obtain $t = u$. □

Proof of Theorem 3.7. By well-founded induction: Assume $e^\dagger(x')$ has been chosen for all $x' \prec_e x$. Then either x is minimal, and $e^\dagger(x)$ can be chosen arbitrarily; or otherwise there are some x', x'', t such that $x' \prec_e x$, $x'' \simeq_e x$, and $e(x'') = (t, x')$. Then choose $e^\dagger(x'') = (t + u, y)$ where $e^\dagger(x') = (u, y)$. By cancellation and Lemma 3.5, necessarily $e^\dagger(x'') = e^\dagger(x)$ for all $x'' \simeq_e x$. □

Proof of Theorem 3.8. Consider the relation \succ_e restricted to each equivalence class C of \sim_e in turn, by Lemma 3.3. Either it is well-founded, such that Theorem 3.7 applies directly; or otherwise the choice of any element x partitions C into the down-set D of x , which is an infinitely descending chain, and its complement $C \setminus D$, which is well-founded. Then fix u, y arbitrarily, and choose $e^\dagger(x') = (u - t, y)$ where by Lemma 3.4, t is uniquely determined by $x \rightsquigarrow_e (t, x')$, for each $x' \in D$. Proceed on $C \setminus D$ as in Theorem 3.7. \square

Proof of Theorem 4.18. Consider a well-formed rule $r = (a, t)$.

The map $\text{step}(t)$ is defined piecewise: glueing points at integer arguments are fixed, by cases (B) and (D). Consider the first pair of adjacent glueing points, at 0 and 1: they are either identical $\vec{0}$, and the map is constant on the whole closed interval $[0, 1]$, by case (A), or they are $\vec{0}$ and \vec{e} , respectively, and a unit curve is spliced continuously in between, by case (C). For further pairs, continuously shifted and turned analogs hold, by cases (E) and (F), respectively. Hence the map is continuous as a whole.

Furthermore, we have $\text{step}(t)(0) = \vec{0}$ by case (B), and $\text{step}(t)(\ell) = a\vec{e}$ by well-formedness and Lemma 4.17. Thus the rescaled map $\text{draw}(r)$ is a unit curve. \square

Proof of Theorem 4.22. By Lemma 2.8, the D -ca-homomorphisms $h : (X, f^\#) \rightarrow (C_1, k)$ can be understood as \mathcal{L}_{wfr} -ca-homomorphisms $h : (X, f) \rightarrow (C_1, \text{draw}_{\text{wfr}})$. That is, they are recursive in the nonterminal slots of a rule. Now proceed by considering the resulting curve pointwise, and case distinction in the workhorse function step :

By the base cases (A), (B) and (D), the inductive definition of h in terms of induction step draw is well-founded, and hence the resulting curve defined uniquely, on a dense subdomain $J \subseteq [0, 1]$; namely on isolated points of recursive interval subdivision by (B) and (D), and on closed subintervals by (A). By the induction hypothesis, the recursively included subcurves are unit curves and hence continuous on J . By finite amount of possible shapes, the extent of subcurves on J is bounded, but they are shrunk recursively by unbounded factors. Thus the resulting curve is totally bounded and hence uniformly continuous on J . By a standard result, a uniformly continuous map on a dense subdomain extends uniquely to a continuous map on its closure, the whole unit domain $[0, 1]$. Clearly, that extension is a unit curve. \square

B Supplementary Figures

This appendix illustrates the construction of fractal curves from L-Systems. Two example systems for very well-known fractals are considered.

- The Koch curve

$$K \xrightarrow{3} K l K r r K l K$$

- The Sierpinski triangle

$$U \xrightarrow{2} l D r U r D l$$

$$D \xrightarrow{2} r U l D l U r$$

where U yields the standard upright triangle, and D its downward mirror image.

The fractals are visualized by equidistantly spaced samples from the domain interval. The unit vector \vec{e} spans the baseline from left to right. Note that *analytically exact* coordinates can be given for each sampled point by recursive symbolic interpretation. Sparse points are connected by dashed lines as a visual aid; these are not valid interpolations of the fractal. Note also that there may be fewer visible points than samples due to non-injectivity.

The figures have been generated by programs written by an author in Haskell and R. The source code is available on request.

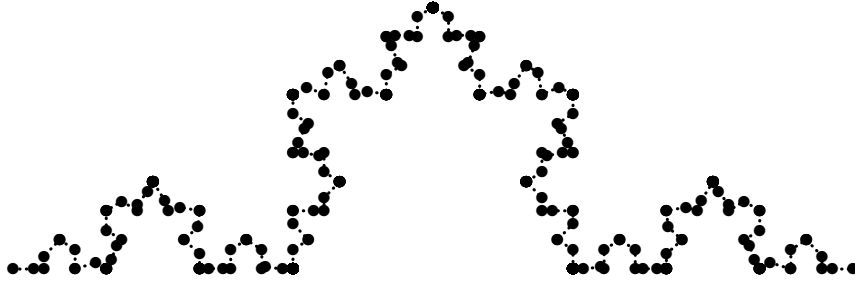


Figure 1: Koch curve $h(K)$, sampled every $1/666$

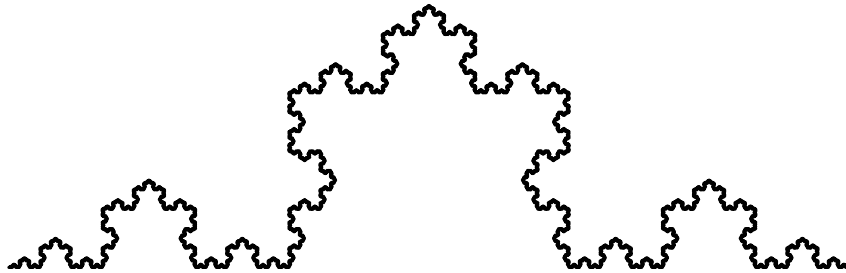


Figure 2: Koch curve $h(K)$, sampled every $1/400000$

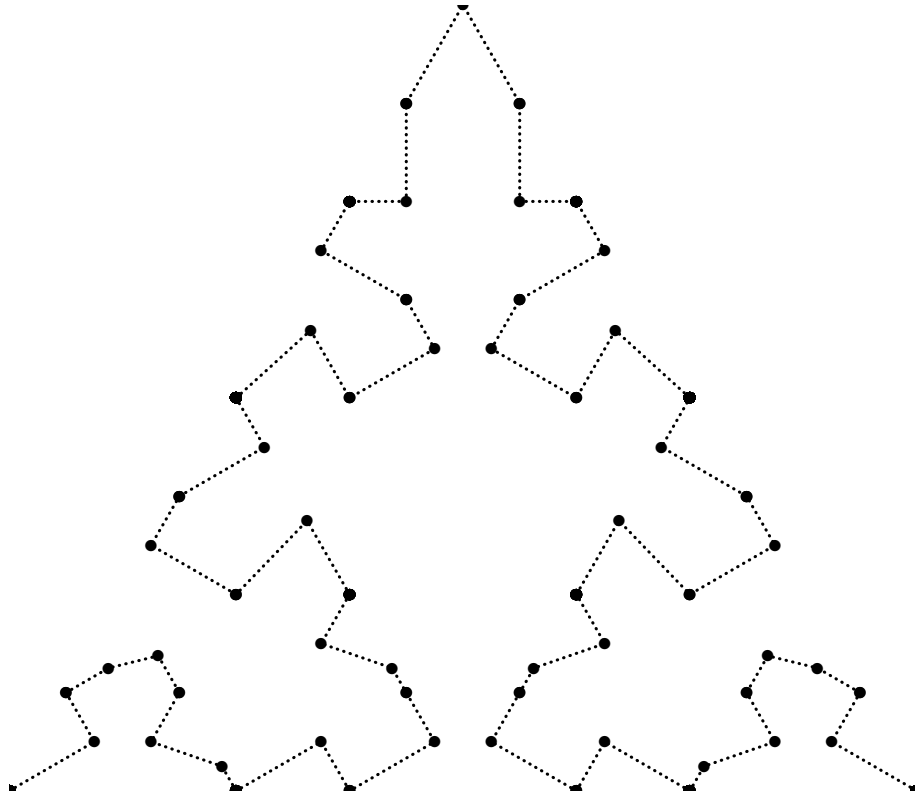


Figure 3: Sierpinski triangle $h(U)$, sampled every $1/666$

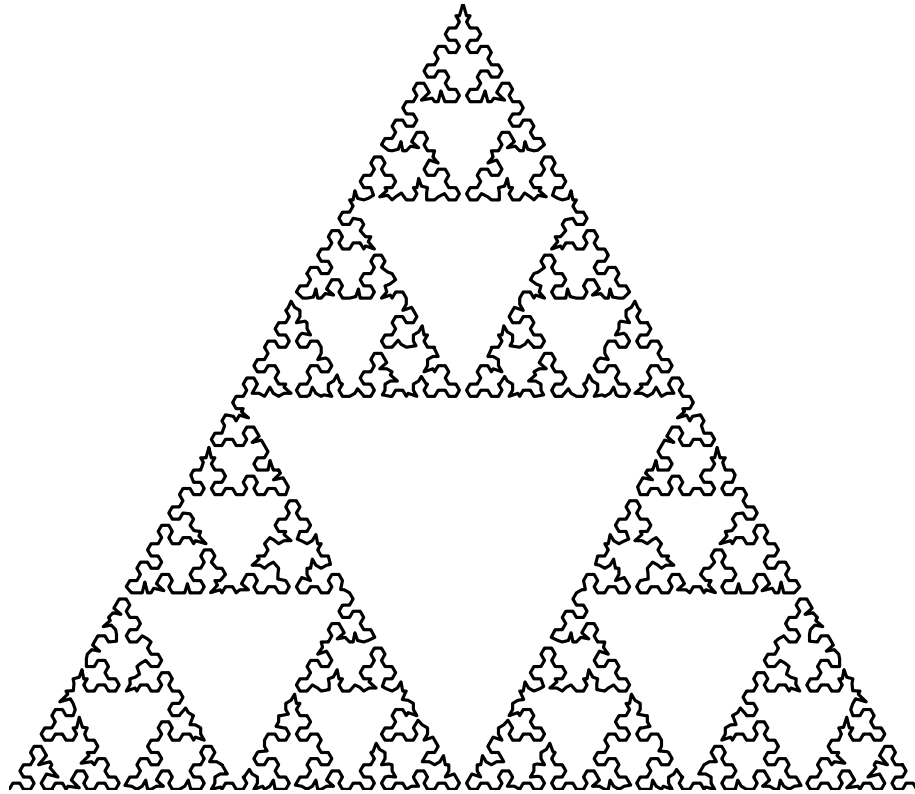


Figure 4: Sierpinski triangle $h(U)$, sampled every $1/400000$